HiPerVison Projects on Shelton Vision's Textile Inspection Systems: Backend web-based interface

Malavika Reghunathan Nair

29 September 2014

An Interim report submitted in fulfilment of the requirements for Course CO7501- Individual Project, Department of Computer Science, University of Leicester.

Supervised by: Dr. Reiko Heckel

Summary:

This project involves working on the Shelton WebSPECTOR surface inspection system which is currently used in various industries to inspect materials for defects. The system uses a combination of vision hardware (lights, cameras and electronics) and software. There are two primary software platforms. The first is the front-end (written in C#) where the operator interface, defect analysis and system co-ordination is done. The second is the back-end (written in C++) where the image processing and defect detection is carried out.

The outline aim of the project is to explore the possibilities of using existing web technologies and services to provide a web-based interface to allow engineers to interrogate the back-end remotely to troubleshoot any faults in identifying defects and adjust parameters.

List of Acronyms:

HTTP Hyper Text Transfer Protocol

REST REpresentational State Transfer

SOAP Simple Object Access Protocol

URI Uniform Resource Identifier

CGI Common Gateway Script

MIME Multi-Purpose Internet Mail Extensions

AJAX Asynchronous JavaScript + XML

IDE   Integrated development environment

JAX-RS Java API for RESTful Services

PHP

JSON

**Table of Contents:**

1. Introduction...............................................................................8

1.1 Overall aim of the project

1.2 Objectives

1.3 Minimum Requirements

1.4 Schedule

1.5 Report Structure

1.6 Reflective Analysis

1. Literature Review

2.1 Introduction

2.2 SOAP, REST and JSON

2.3 Client-Side scripting techniques

2.3.1 Javascript , JQuery and AJAX

2.3.2 PHP and AJAX

2.3.3 ASP.NET and AJAX

2.4 Jersey Framework

1. Environment Set up

3.1 Database – MSSQL

3.2 IDE - Eclipse

3.3 Server – Oracle Weblogic

3.4 Obstacles faced

4. Implementation of a RESTful web service

5. Implementation of client using Javascript, JQuery and Ajax

6. Implementation of client using PHP

7. Implementation of client using ASP.NET and JQuery

8. Evaluation of the results and comparison

9. Conclusion and feature works

Bibliography

Appendix A- Schedule

Appendix B- Installation

**List of Figures:**

Figure 1a: Schedule used for the project with major tasks and the

time scale.

Figure 1b: Figure 1b: Continuation of Figure 1a

Figure 2: Web Technology

Figure 3: Communication between a user and web-based database

Figure 4: RESTful WebService

Figure 5: Test Database Specification

Figure 6:Java RESTful Service Code snippet

**List of Tables:**

Table 1: Performance result of SOAP and REST

Chapter 1

1. **Introduction**
   1. **Overall aim of the project**

‘Shelton Vision’ company based in UK is aiming to have a back-end web based interface to allow its engineers to interrogate the backend remotely to troubleshoot any faults in identifying defects and adjust parameters. Right now, they have a desktop based application to do the job.

The aim of this project is to do a feasibility study of current technologies available and decide on the best technology to implement the client-side web application with the features as of the existing desktop application. Most of the features are repeating and therefore the most important features to implement at client-side are decided after discussing with the client.

By implementing the client side with different technologies and comparing the quality of web-based interface with the desktop application, it will be possible to conclude which will be the best option out of it.

* 1. **Objectives**

The project assesses the feasibility of implementing a Web-based interface for the engineers to access the information linked to a product database. The information the engineer sees is a mixture of material images, process graphs and other displays. The existing system works under the Microsoft windows XP operating system and a MSSQL database. It has been developed over 15 years with a mixture of Visual Basic 6, C and assembler libraries, SQL procedures, and WiT.

The project starts with a background research of available client-side scripting techniques. The project is supervised in cooperation with the client, who provides the requirements and technical support based on their understanding of their needs.

The main objective of this project is to obtain a basic understanding of client-server programming, obtain a good grasp of client-side scripting techniques and finally draw a conclusion to use a particular scripting to achieve quality results.

After discussing with the client (Shelton Machines Ltd) for whom the project is done for, the functionalities to be implemented have been decided as follows:

* Display image at client side which is at a storage and the path to image is stored in the database
* Update parameters in the database
* Display graph at client side from vector data

**1.3 Minimum Requirements**

• Successful implementation of a database to store the test data (MSSQL preferred as the client uses that)

• Successful implementation of an IDE (Eclipse IDE for Java EE developers)

• Successful implementation of a web server and servlet container (Oracle Weblogic)

• A framework for developing web services (Jersey framework)

• Evaluate the performance and scalability by comparing the functionality with the existing desktop application

**1.4 Schedule**

A Gantt chart was prepared to manage the allocation time of the project tasks. The schedule started from June 1st 2014 and each column in Gantt chart (Figure 1a and 1b) represents a week-ending. There were regular meeting with the supervisor and the company CTO to decide on the major functionalities and helped to decide on how to go about the solution.

As the solution was not known for this project, it aims at a feasibility study of available client side scripting techniques to figure out the best technology that suits the needs of the client. Sufficient time was allocated to do the background research of available technologies and to set up the environment.

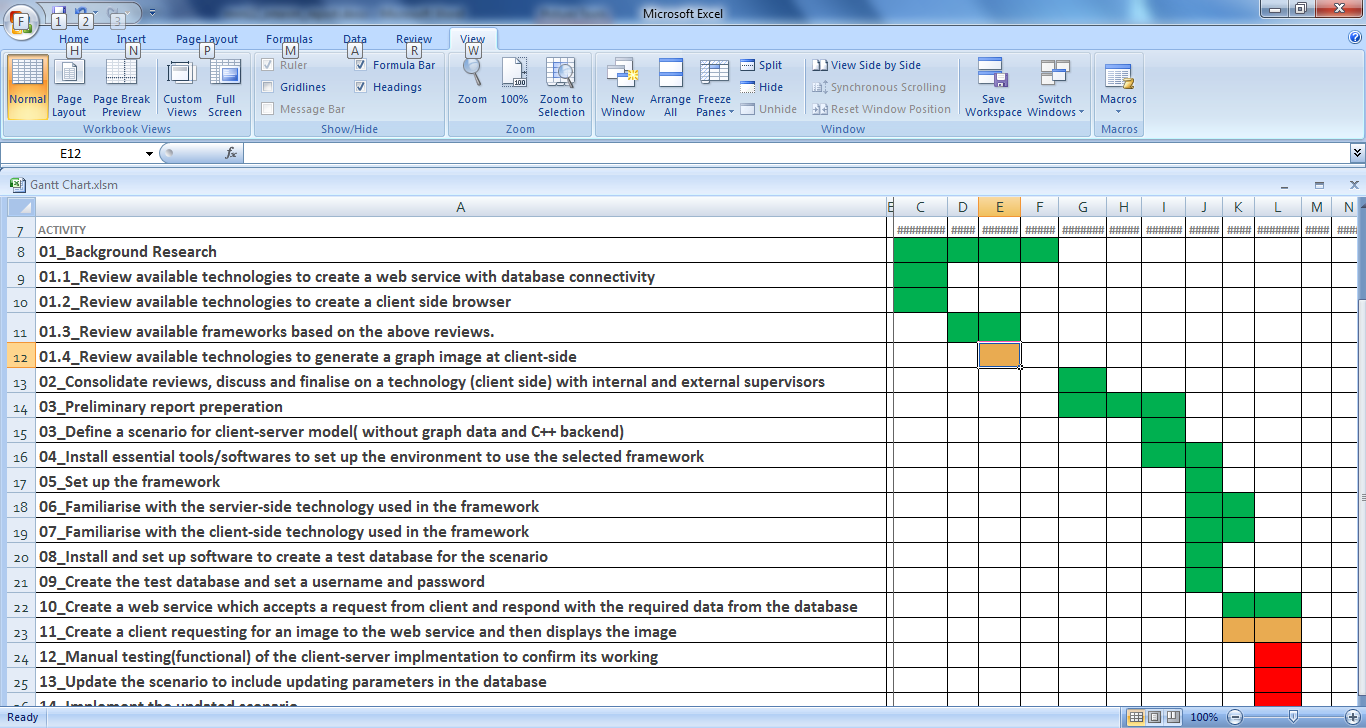


Figure 1a: Schedule used for the project with major tasks and the time scale.

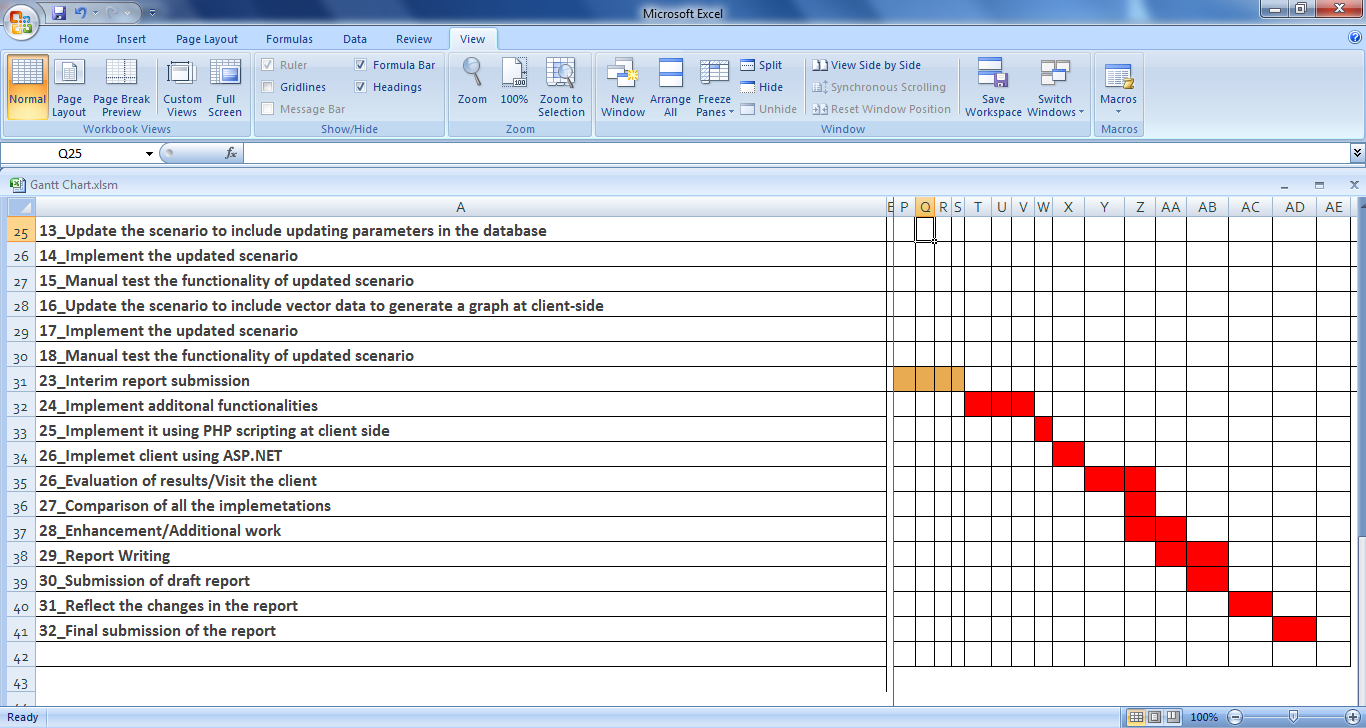


Figure 1b: Continuation of Figure 1a

**1.5 Report Structure**

The following is the structure of the report.

* Chapter 2 describes the background research on client-server architecture, basics of web technology, specifications used for web services, technologies available for client-side scripting and a brief description of a framework that can be used to build web services.
* Chapter 3 describes the setting up of environment essential to perform a client-server task. Basically, the IDE, database, Operating system, Server etc and the obstacles faced during the progress
* Chapter 4 describes the implementation of a RESTful web service
* Chapter 5 describes the implementation of client and client using JavaScript, JQuery and AJAX
* Chapter 6 describes the implementation of client using PHP and AJAX
* Chapter 7 describes the implementation of client using ASP.NET and JQuery
* Chapter 8 evaluates the results to check the performance, to do a comparison of all the above implementations in order to decide on the best technology that suits the project
* Chapter 9 concludes the overall impact of the project

**1.6 Reflective Analysis**

According to the initial plan, a web service which is able to communicate to C++ applications was a requirement. But it was later removed and decided to give more focus to the Client-side browser application.

Chapter 2

1. **Literature Review**

**----------------------------------------------------------------------------------------------------------------**

* 1. **Introduction**

The web is a distributed, dynamic and large information repository [3]. Communication over the web or internet can be broken down to two interested parties: *Clients and Servers*. The machine providing services are servers. Clients are the machine used to connect to those services [10]. *Services* are self-contained modules-deployed over standard middleware platforms- that can be described, published, located, orchestrated and programmed using technologies over a network [4].

A web browser is the web client which acts on behalf of the user. The browser contacts the web server and sends a request for the information and receives the information and displays it on the user’s computer [10]. Fig 2.1a shows how a basic web technology works.

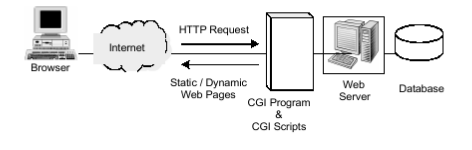


Figure 2: Web Technology [10]

The figure 2.1b illustrates the steps for a web page to access the database. A web browser cannot directly access a database. Most of the cases, browsers are a program running on the web server which acts as an intermediary to the database [10].

When the user hits a URL or clicks a submit button on the web page, the browser sends the request to the web server, which passes it to the Common Gateway Script (CGI). The CGI loads a library which sends the SQL commands to the SQL database server. The database server then executes the query and sends the result to the CGI script. The CGI script generates an HTML document and writes it to the web server. The web server sends the HTML page back to the remote user [10].

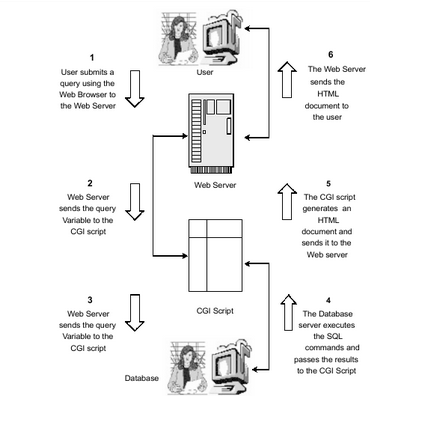


Figure 3: Communication between a user and web-based database [10]

The main goal of web service is to exchange information among applications in a standard way [9]. Two most widely used approaches for web service development are SOAP and REST (Representational State Transfer).REST has been accepted widely as a simpler alternative to SOAP and WSDL based web services [8].

* 1. **SOAP, REST and JSON**

SOAP, REST and JSON are three specifications of web services.

**SOAP** defines a communication protocol for web services. WSDL enables service providers to describe their applications. UDDI offers a registry service that allows advertisement and discovery of web services [3]. XML is used to define Simple Object Access Protocol (SOAP).

**REST** (Representational Stat Transfer) defines a set of architectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages [8].Basically, web services are viewed as resources and can be identified by their URLs. Client and server communicate by sending and receiving representation of resources. Resources are commonly represented using JSON rather than XML because it is more compact than XML and it can be used with almost all programming languages including JavaScript [5]. JAX-RS uses annotations to simplify RESTful web service development. By adding annotations, we can define resources and can define the operations or actions to be performed on those resources.

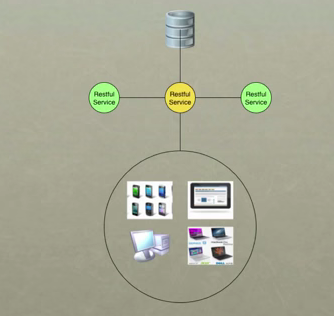
****

Figure 4: RESTful Web Service

**JSON** (JavaScript Object Notation)is an open standard format that uses a non-strict subset of JavaScript. Information is exchanged using data objects in the form of attribute-value pair. The MIME type for JSON text data is “application/json”.

JSON is much simpler than XML and is a better data exchange format.

JSON Sample:

{

“id”: 1,

“name”: “Dave”,

“city”: “London”

“gender” : { “type” : “male”

},

“phone number” :{ “type” : “work”,

“number” : “000 007 131”

}

}

We could use a combination of web service specifications in order to obtain a better performance.

**Advantages of REST over SOAP**

Web services performance is an important factor. SOAP communications causes network traffic, higher latency and processing delays. To overcome this limitations the RESTful architecture is used. REST is a lightweight, easy and better alternative for the SOAP [9]. Table 1 illustrates a performance comparison of SOAP and REST in terms of message size and time.

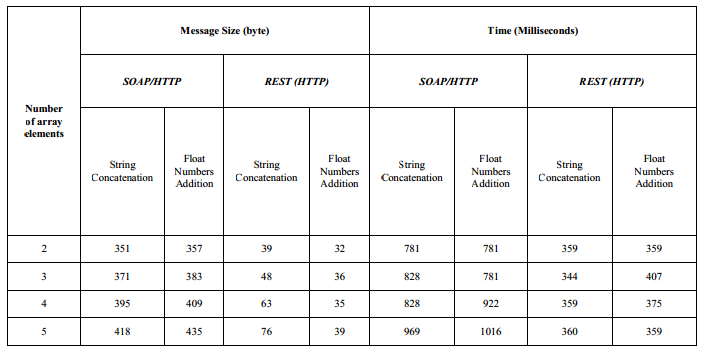


Table 1: Performance result of SOAP and REST [11]

* 1. **Client-Side scripting techniques**

**2.3.1 Javascript, JQUERY and AJAX**

**AJAX** is the ability of a webpage to send and retrieve data asynchronously from a server, without interfering with the display and actions on the webpage. JSON is mostly used in AJAX instead of XML.

**JQUERY** is a free, open-source and cross-platform JavaScript library used to simplify the client-side scripting of HTML. It has become very popular today and mostly used to develop dynamic web pages. It is used to develop Ajax applications. Using JQuery library eliminates cross-browser incompatibilities.

2.3.2 PHP and AJAX

[to be done…]

2.3.3 ASP.NET and JQuery

[to be done..]

* 1. **Jersey Framework**

“Jersey RESTful Web Services framework is an open source, production quality JAX-RS reference implementation for building RESTful Web Services in Java” [6].Jersey produces and consumes RESTful web services. In Jersey framework, the following methods are used which are defined in the HTTP/1.1 protocol.

* @GET is used to retrieve data or perform a query on a resource. The data returned from the web service is a representation of the requested resource [7]. This is the most-used, read only , public access method
* @POST is used to create a new resource. The web service may respond with data or status indicating success or failure [7]. With HTTPS, we can protect data
* @PUT is used to update existing resources or data [7]. But it can also be used for inserting or adding data.
* @DELETE is used to remove a resource or data [7].
* @HEAD used to return meta-data of the resource

**Sample code:**

@Path("/service/")

**public** **class** Rest\_Service {

@Path("/database/imagepath/\*")

@GET // HTTP verb is required to access this method

@Produces(MediaType.*TEXT\_HTML*)

**public** String getImagePath() **throws** Exception {

// code goes here

}

}

* 1. **Conclusion**

Based on the background research, it was decided to create a Java RESTful service to test the different client-side scripting techniques.

Chapter 3

**3.** **Environment Set Up**

**----------------------------------------------------------------------------------------------------------------**

* 1. **Database - MSSQL**

The database used for this project is MSSQL 2003 as recommended by the client (Shelton Machines Ltd). A test database was created whose specifications are shown in figure5.

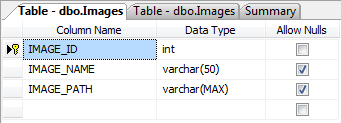
****

Figure 5: Test Database Specification

The images are of size 4MB.For time being, the images are stored in the C drive of the machine.

* 1. **IDE - Eclipse**

Using Eclipse IDE for development of Java RESTful service simplifies the task significantly and offers several tools. Eclipse Indigo which comes along with Oracle Weblogic 12c installation in used in this project.

With a new Dynamic Web Project created in the eclipse IDE for Java EE developers, it is possible to create both service and client in the same package through a web design view. Still the coding needs to be done manually. Also it makes the deployment of the code to the server easy.

In order to make the Java RESTful service work, the following Java library files were added:

* Jar file from <http://Jersey.java.net/>
* Jar file from <http://Jackson.codehaus.org/>
  1. **Server – Oracle Weblogic**

This project makes use of a Java application Server to deploy the service. “Oracle WebLogic Server 12c is the industry's best application server for building and deploying enterprise [Java EE](http://www.oracle.com/technetwork/java/javaee/overview/index.html) applications with support for new features for lowering cost of operations, improving performance, enhancing scalability and supporting the [Oracle Applications](http://www.oracle.com/us/products/applications/index.html?ssSourceSiteId=otnen) portfolio “[12].

Oracle Weblogic 12c was installed with the default settings and Oracle WebLogic 12.1.1 is used for all work in this project. It operates by default on port 7001. In this project, service is deployed to it using Eclipse IDE. A new domain is created with a username and password so that we can login to admin console of the server. In order to connect the server with the database, a data source needs to be created. An SQL data source is created from admin console of the server and a connection is established to the database created in MSSQL.

* 1. **Obstacles faced during set up**

Problem: Installation of MSSQL 2003 in Windows 8 Operating System was

unsuccessful. MSSQL 2003 is the database currently using by the

client (Shelton Machines Ltd)

Solved By: After discussing with the client (Shelton Machines Ltd), ordered

Windows 7 Home Premium OS online as it was not available in

shops. Following the installation of Windows 7, MSSQL 2003

was successfully installed.

Chapter 4

**4.** **Implementation of a RESTful web service**

**----------------------------------------------------------------------------------------------------------------**

The REST Service is written in Java and runs on Oracle WebLogic Server.The service is written in the template provided by Eclipse when a new dynamic web project is created. It loads the necessary libraries and provides options such as configuring the server. Oracle WebLogic server is then added as a new server to the project. The rest of the coding for the service should be done manually.

Figure 6 shows a code snippet of the RESTful web service. As seen in line 41, the HTTP verb “POST” is used to invoke the service method.

The method retrieves the SQLDataSource of WebLogic which was created as explained in section 3.3 and returns a connection. The code snippet is showed in Figure 7. The method response is the image path to the client as a JSON object. Code snippet of the method doing the SQL queries is shown in Figure 8.

Three packages were used to separate the RESTful service from SQL Logic and JSON object mapping. The advantage of this is that if the REST service needs changing, there is no risk of breaking the SQL connection and logging packages. The structure of the packages is shown in figure 9.

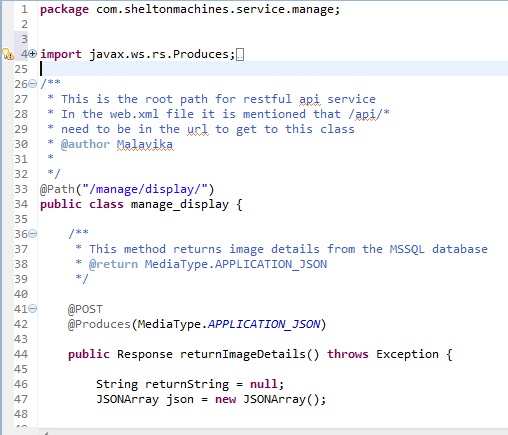


Figure 6:Code snippet of Java RESTful Service Code snippet

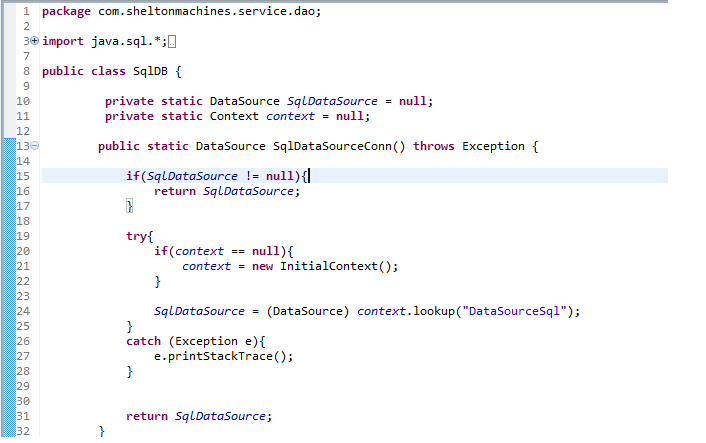
****

Figure 7: Code snippet of the Java method to retrieve the DataSource created in WebLogic

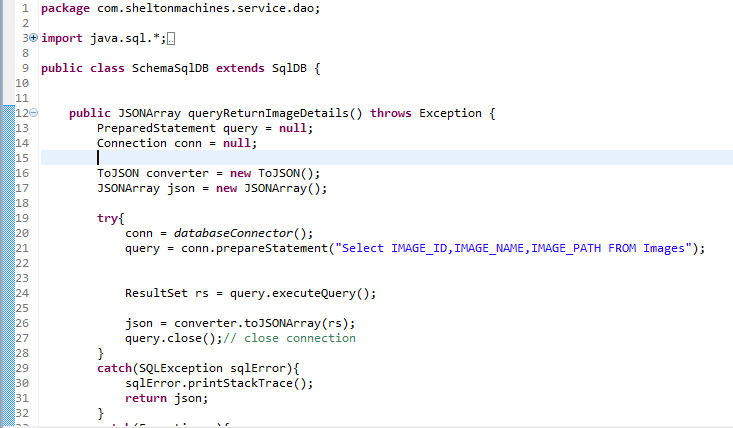
****

Figure 8: Code snippet of the method which performs the SQL query to the database

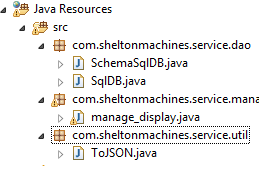
****

Figure 9: REST Service Package

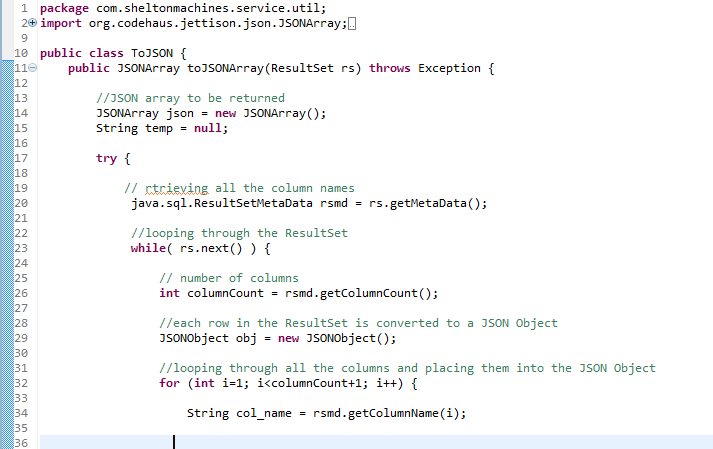
****

Figure 7: Code snippet for converting the database ResultSet to JSON Object

Chapter 5

**5.** **Implementation of client using Javascript, JQuery and Ajax**

**----------------------------------------------------------------------------------------------------------------**

The client is written in Javascript, JQuery and Ajax and the client access the service on the WebLogic server.

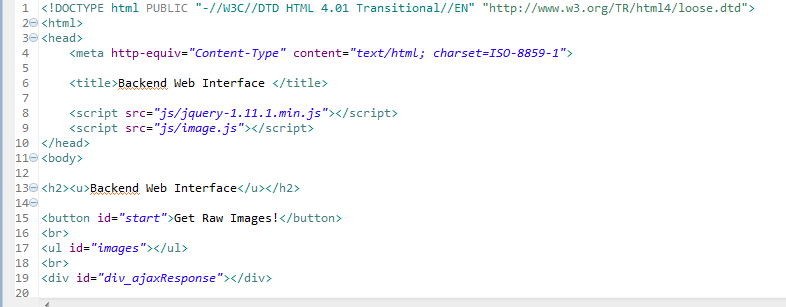
****

Figure 8 : Code snippet of the html page for the client

****

Figure 9: Code snippet of JQuery + Ajax consuming the REST service

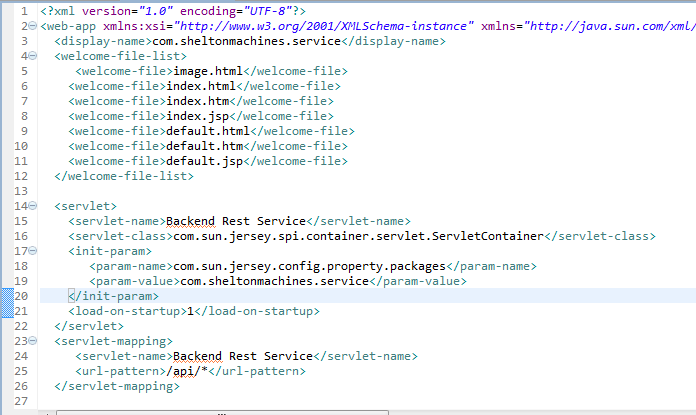
****

Figure 10:

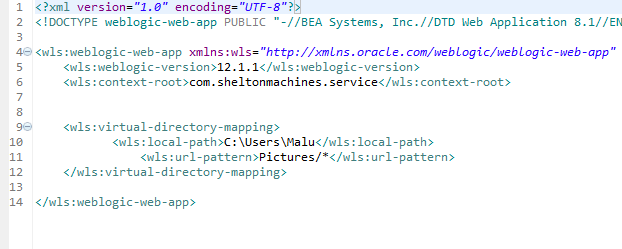
****

Figure 11:

**Virtual directory mapping**

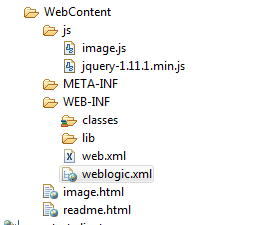
****

Figure 12: Client Package

****

Figure 13: The client page

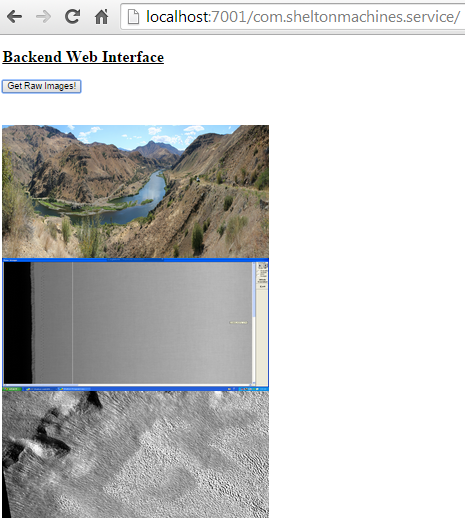
****

Figure 14: Result page

**References:**

[1] “JQuery UI 1.8. The User Interface Library for JQuery”,by Dan Wellman

[2]”Apache CXF Web Service Development-Develop and deploy SOAP and RESTful web services”, by Naveen Balani and Rajeev Hathi,PACKT PUBLISHING, BIRMINGHAM-MUMBAI

[3]”Foundations for Efficient Web service Selection”by Qi Yu, Athman Bouguettaya, SPRINGER US, 2010

[4]”Web Services & SOA Principles and Technology”, by Michael.P.Papazoglou, SECOND EDITION,PEARSON

[5]Part 1: Introduction to Jersey—a Standard, Open Source REST Implementation [online], Accessed date [27/09/2014]

Available at: <http://www.oracle.com/technetwork/articles/javaee/jersey-part1-159891.html>

[6] “RESTful Web Services with Jersey” by [Scott Leberknight](http://www.slideshare.net/scottleber?utm_campaign=profiletracking&utm_medium=sssite&utm_source=ssslideview), Software Developer at Near Infinity Corporation, Jun 07, 2014

[7] RESTful Web Services[online], [Accessed date: [27/09/2014]

Available at: <http://www.oracle.com/technetwork/articles/javase/index-137171.html>

[8] RESTfulWeb services: The basics [online], [Accessed date: 14/07/2014]

Abailable at: <http://www.ibm.com/developerworks/webservices/library/ws-restful/>

[9] Web Services based on SOAP and REST Principles [online], [Accessed date: 25/09/2014]

Available at : <http://www.ijsrp.org/research-paper-0513/ijsrp-p17115.pdf>

[10] “Introduction to Client Server Computing” by Yadav, Subash Chandra, Singh, Sanjay Kumar, New Age International, 2009

[11] Hatem Hamad, Motaz Saad, and Ramzi Abed “Performance Evaluation of RESTful Web Services for Mobile Devices” Computer Engineering Department, Islamic University of Gaza, Palestine International Arab Journal of e-Technology, Vol. 1, No. 3, January 2010.

## [12] Oracle WebLogic Server [online], [Accessed date: [27/09/2014]

Available at: <http://www.oracle.com/technetwork/middleware/weblogic/overview/index.html>